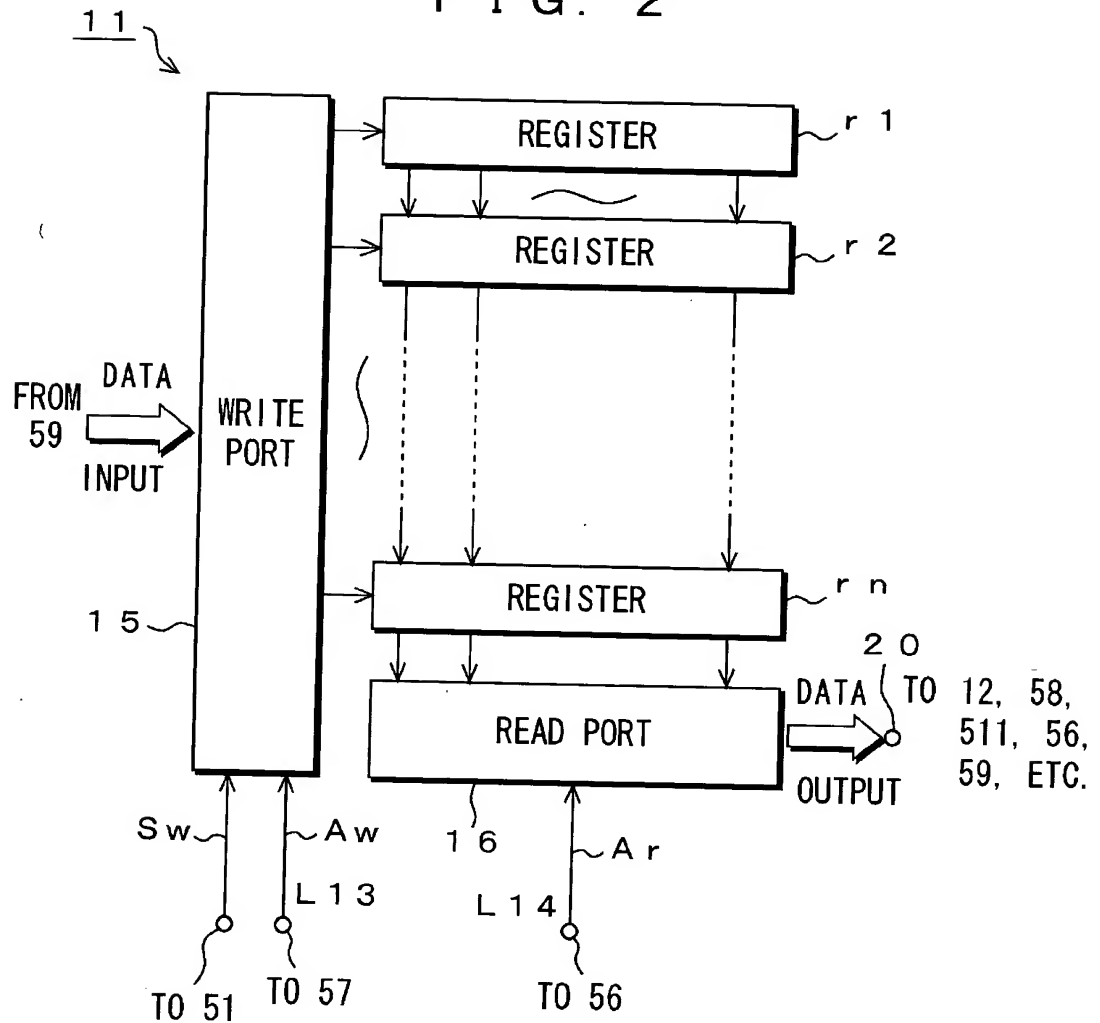


2 / 2 5

F I G . 2



3 / 2 5

FIG. 3A
load, add, OR
cmp INSTRUCTION

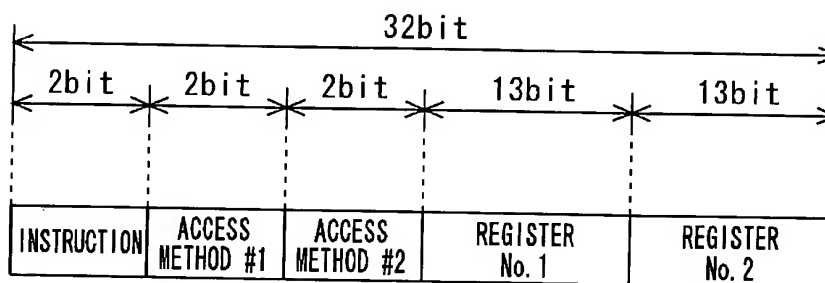


FIG. 3B
jump
INSTRUCTION

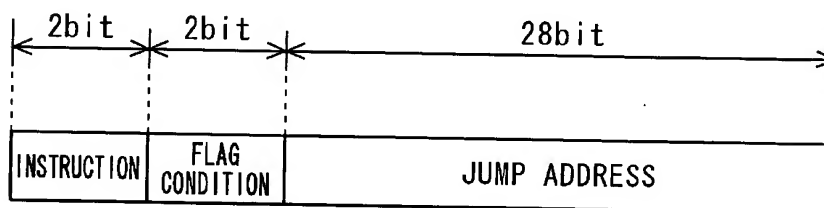


FIG. 3C

INSTRUCTION	
0	load
1	add
2	cmp
3	jump

FIG. 3D

ACCESS METHOD	
0	REGISTER, DIRECTLY
1	REGISTER-TO-REGISTER
2	REGISTER-TO-OUTSIDE
3	NOT USED

FIG. 3E

FLAG CONDITION	
0	UNCONDITIONAL
1	zero flag
2	non-zero flag
3	NOT USED

4 / 2 5

F I G . 4

N o .	EXPRESSION BY MNEMONICS		EXPRESSION BY MACHINE LANGUAGE
# I 1	LOOP :	add[r10], r11	5001400Bh
# I 2		add r10, r11	4001400Bh
# I 3		cmp r10, r12	8001400Ch
# I 4		jump nz, LOOP	E0000000h

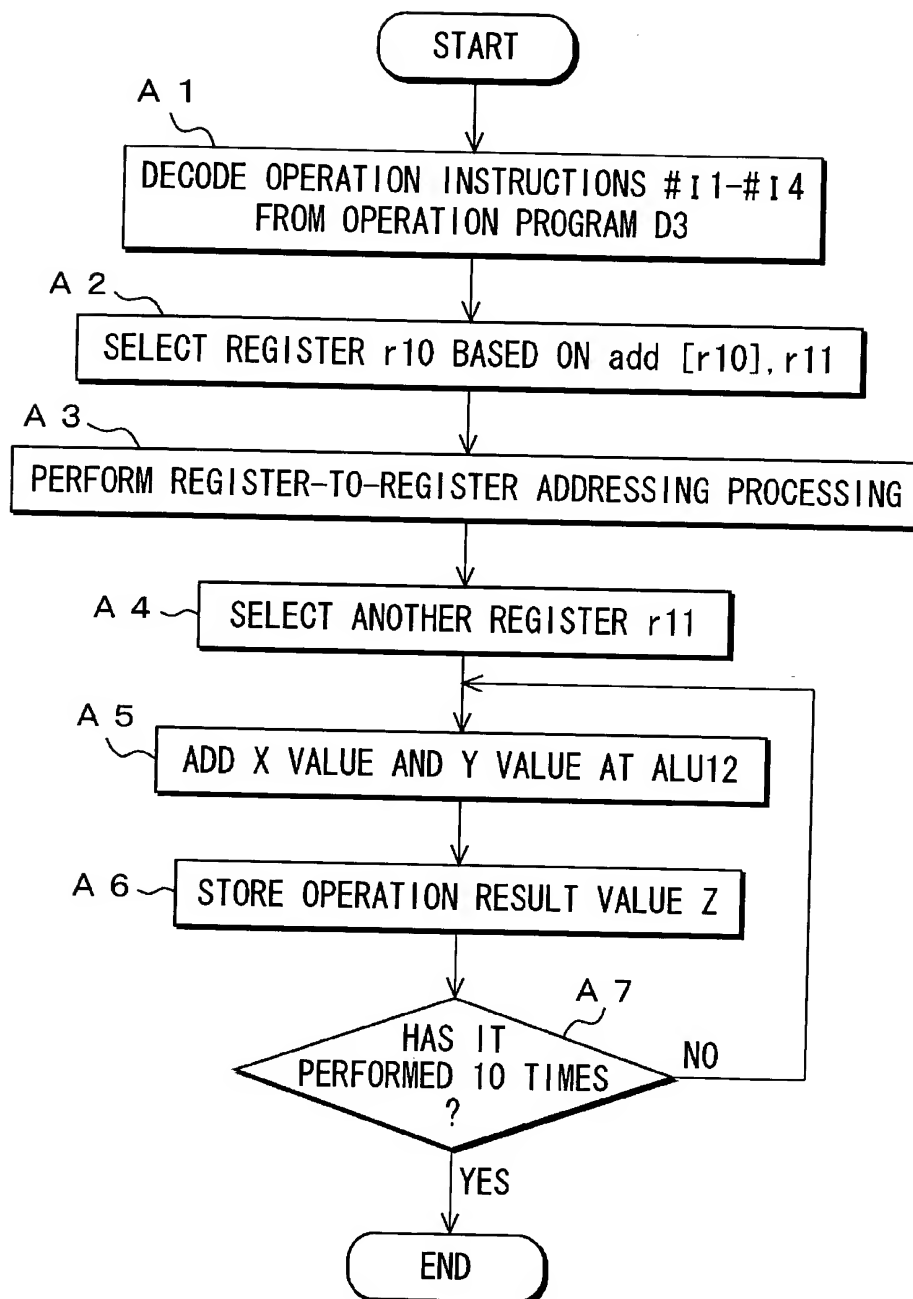
F I G . 5

REGISTER No.		VALUE	
r 0		0	
r 1		0	
r 2		0	
r 3		0	
r 4		0	
r 5		0	
r 6		0	
r 7		0	
r 8		0	
r 9		0	
r 10		0	← No. OF CURRENT ARRAY
r 11		1	← VALUE TO BE ADDED
r 12		10	← END VALUE

11 ~

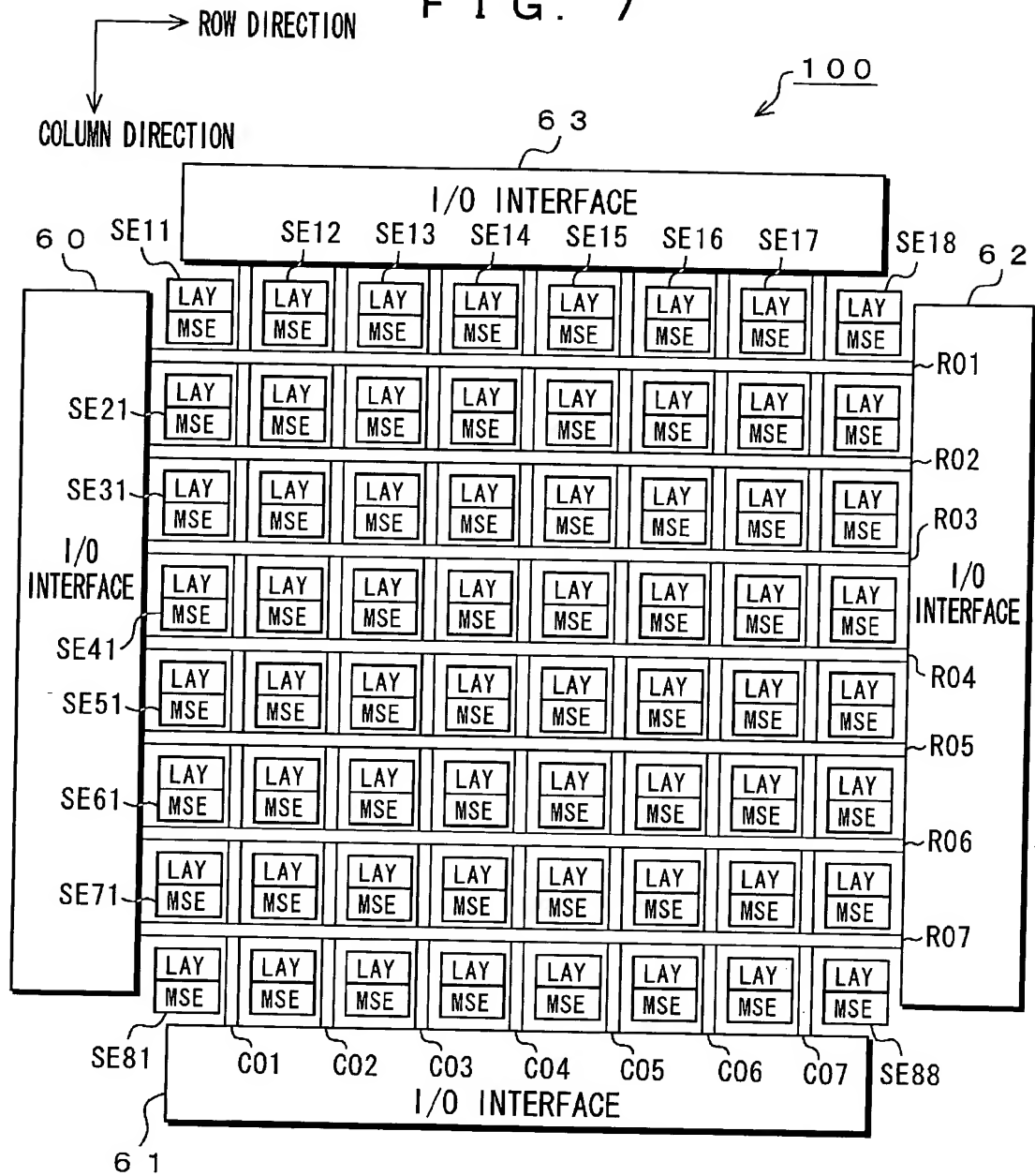
5 / 25

FIG. 6



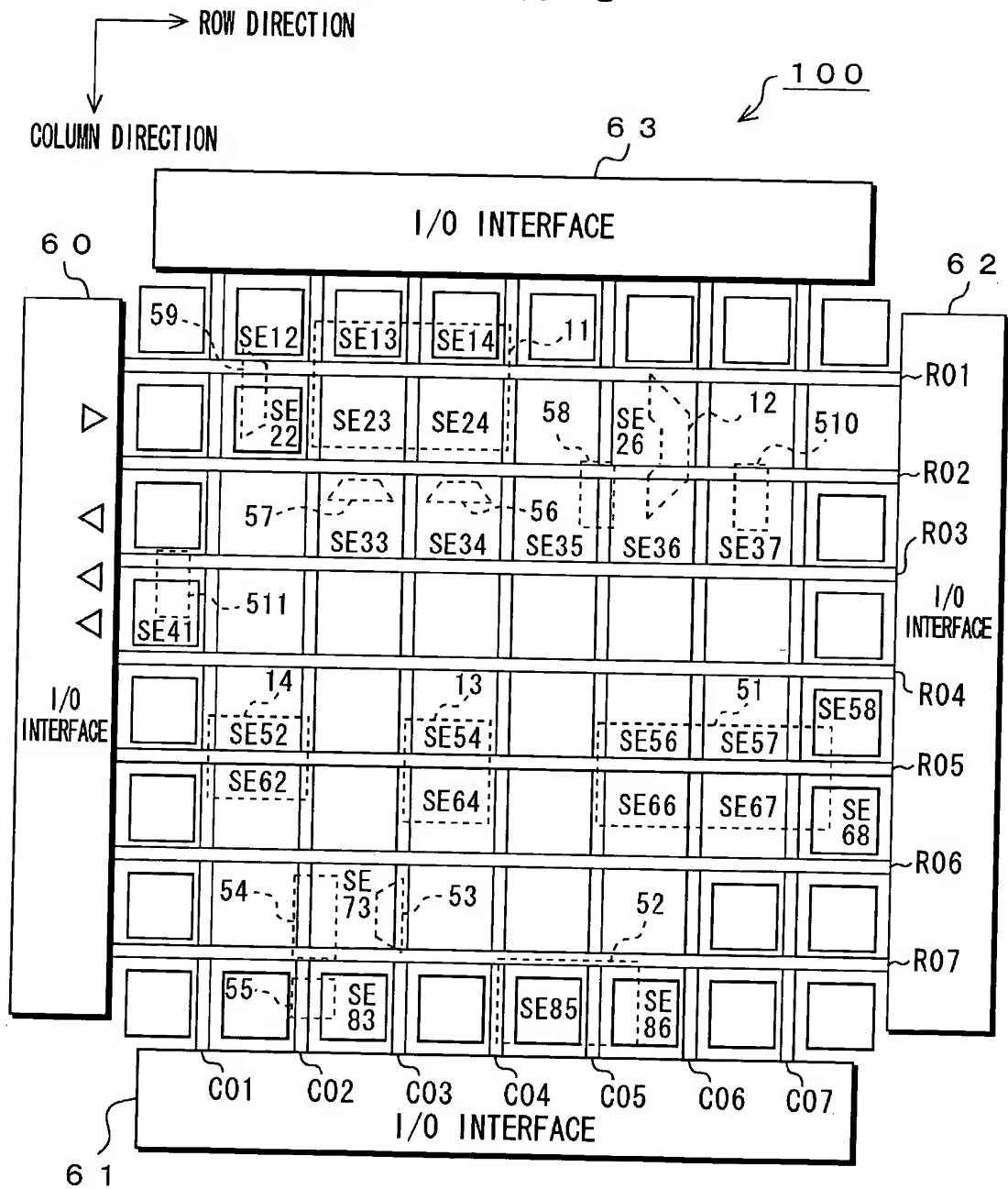
6 / 2 5

F I G . 7



7/25

FIG. 8



8 / 25

FIG. 9

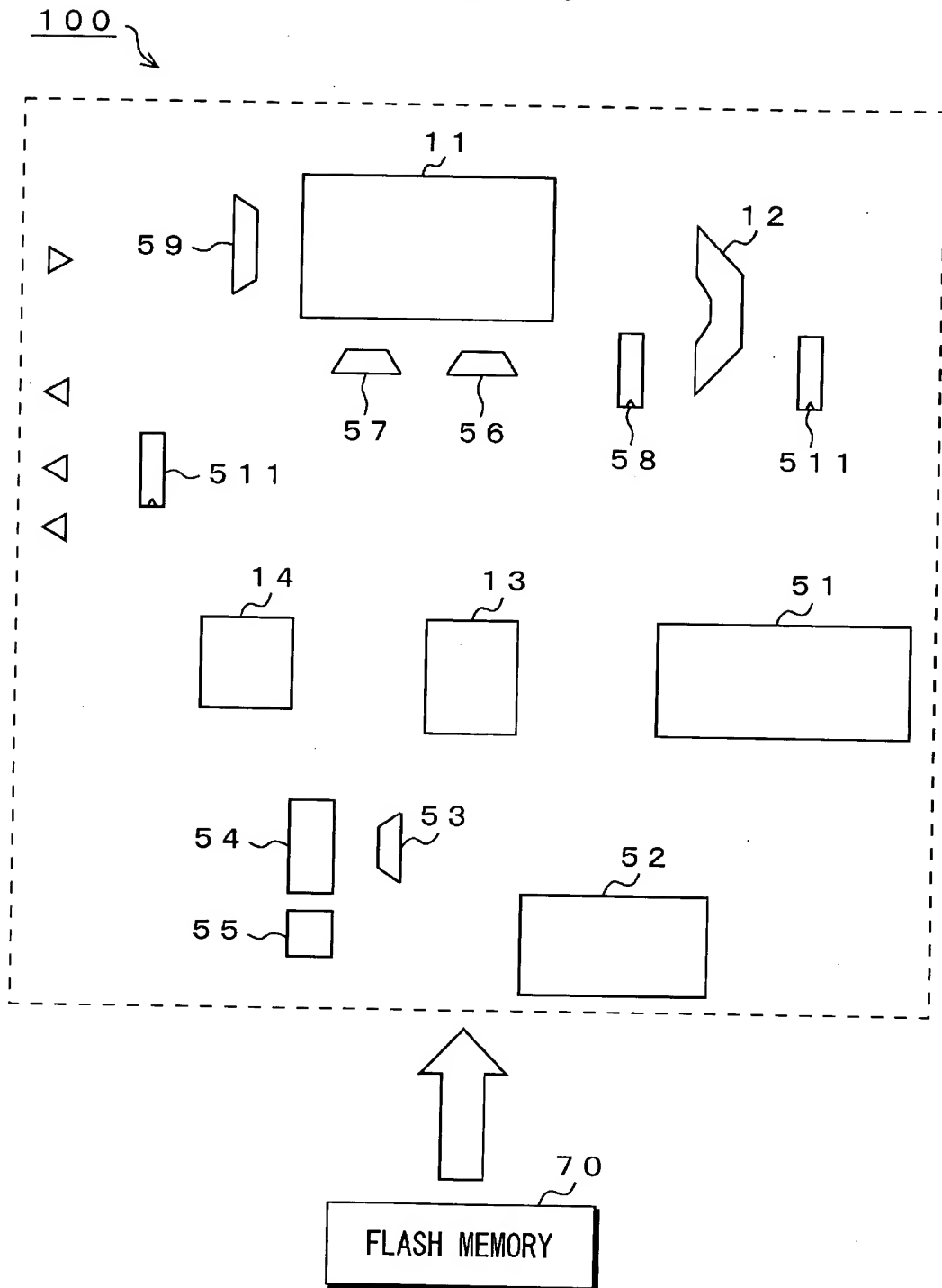


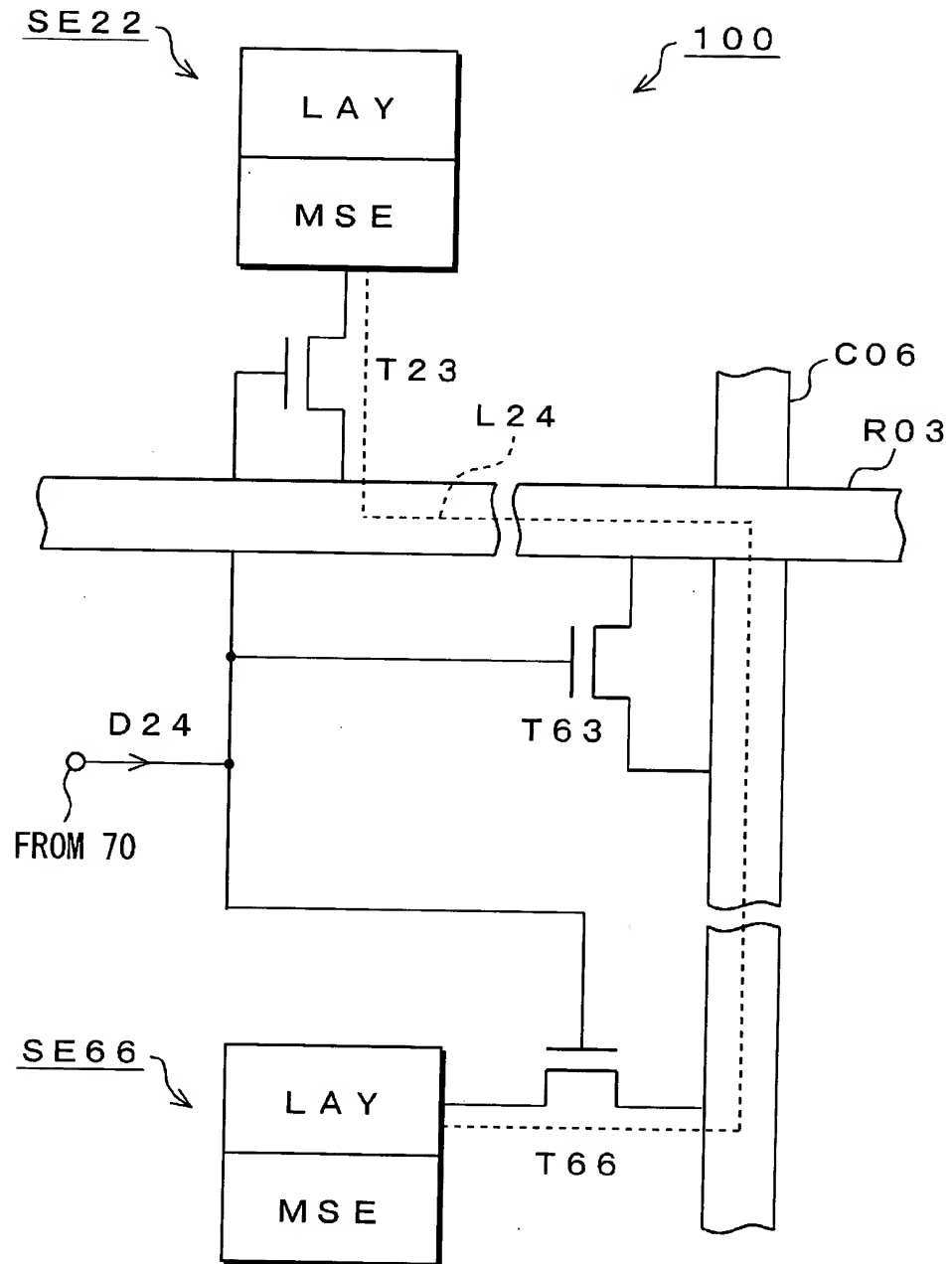
FIG. 10

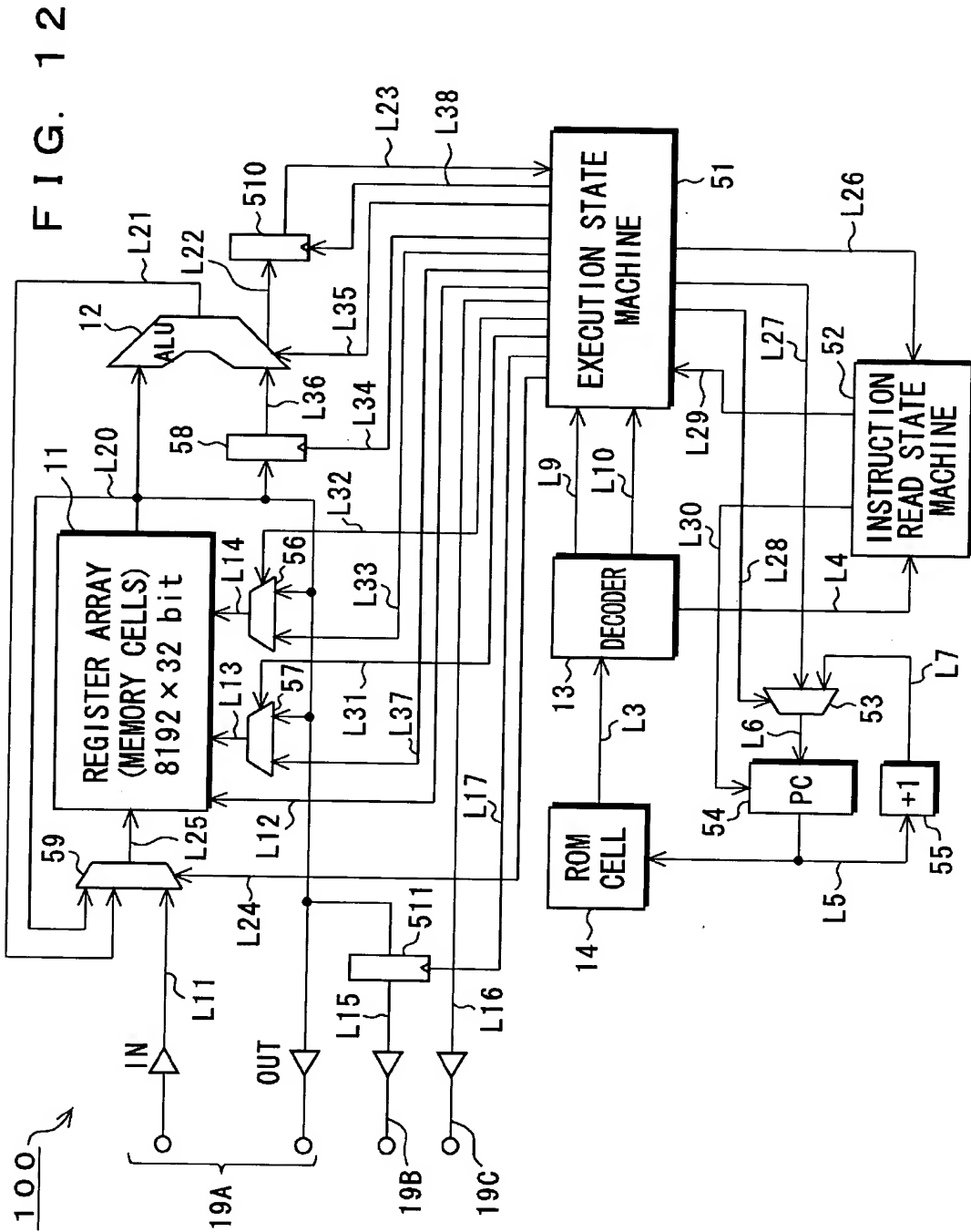
70 ↘

WIRING INFORMATION ITEM	SIGNAL LINE	USE	WIRING INFORMATION ITEM	SIGNAL LINE	USE
D 1	NOT USED		D 2 1	L 2 1	OPERATION RESULT VALUE Z
D 2	NOT USED		D 2 2	L 2 2	COINCIDENCE DETECTION SIGNAL S22
D 3	L 3	OPERATION PROGRAM AP	D 2 3	L 2 3	FLAG CONDITION SIGNAL S23
D 4	L 4	INSTRUCTION CONTROL SIGNAL S4	D 2 4	L 2 4	SELECTION CONTROL SIGNAL S24
D 5	L 5	COUNT OUTPUT SIGNAL S5	D 2 5	L 2 5	DATA, Z, X
D 6	L 6	SELECTOR OUTPUT S7, S27	D 2 6	L 2 6	EXECUTION END SIGNAL S26
D 7	L 7	INCREMENT OUTPUT SIGNAL S7	D 2 7	L 2 7	BRANCH-OFF CONTROL SIGNAL S27
D 8	NOT USED		D 2 8	L 2 8	SELECTION CONTROL SIGNAL S28
D 9	L 9	INSTRUCTION SIGNAL S9	D 2 9	L 2 9	INSTRUCTION EXECUTION START SIGNAL S29
D 10	L 10	EACH AUGMENT SIGNAL S10	D 3 0	L 3 0	COUNT CONTROL SIGNAL S30
D 11	L 11	DATA	D 3 1	L 3 1	SELECTION CONTROL SIGNAL S31
D 12	L 12	WRITE CONTROL SIGNAL Sw	D 3 2	L 3 2	SELECTION CONTROL SIGNAL S32
D 13	L 13	WRITE ADDRESS Aw	D 3 3	L 3 3	READ ADDRESS Ar
D 14	L 14	READ ADDRESS Ar	D 3 4	L 3 4	LATCH CONTROL SIGNAL S34
D 15	L 15	EXTERNAL ADDRESS	D 3 5	L 3 5	ALU CONTROL SIGNAL S35
D 16	L 16	EXTERNAL CONTROL SIGNAL S16	D 3 6	L 3 6	X VALUE
D 17	L 17	LATCH CONTROL SIGNAL S17	D 3 7	L 3 7	WRITE ADDRESS Aw
D 18	NOT USED		D 3 8	L 3 8	LATCH CONTROL SIGNAL S38
D 19	NOT USED		•		
D 20	L 20	DATA, X, Y	•		
			•		

10/25

FIG. 11





12/25

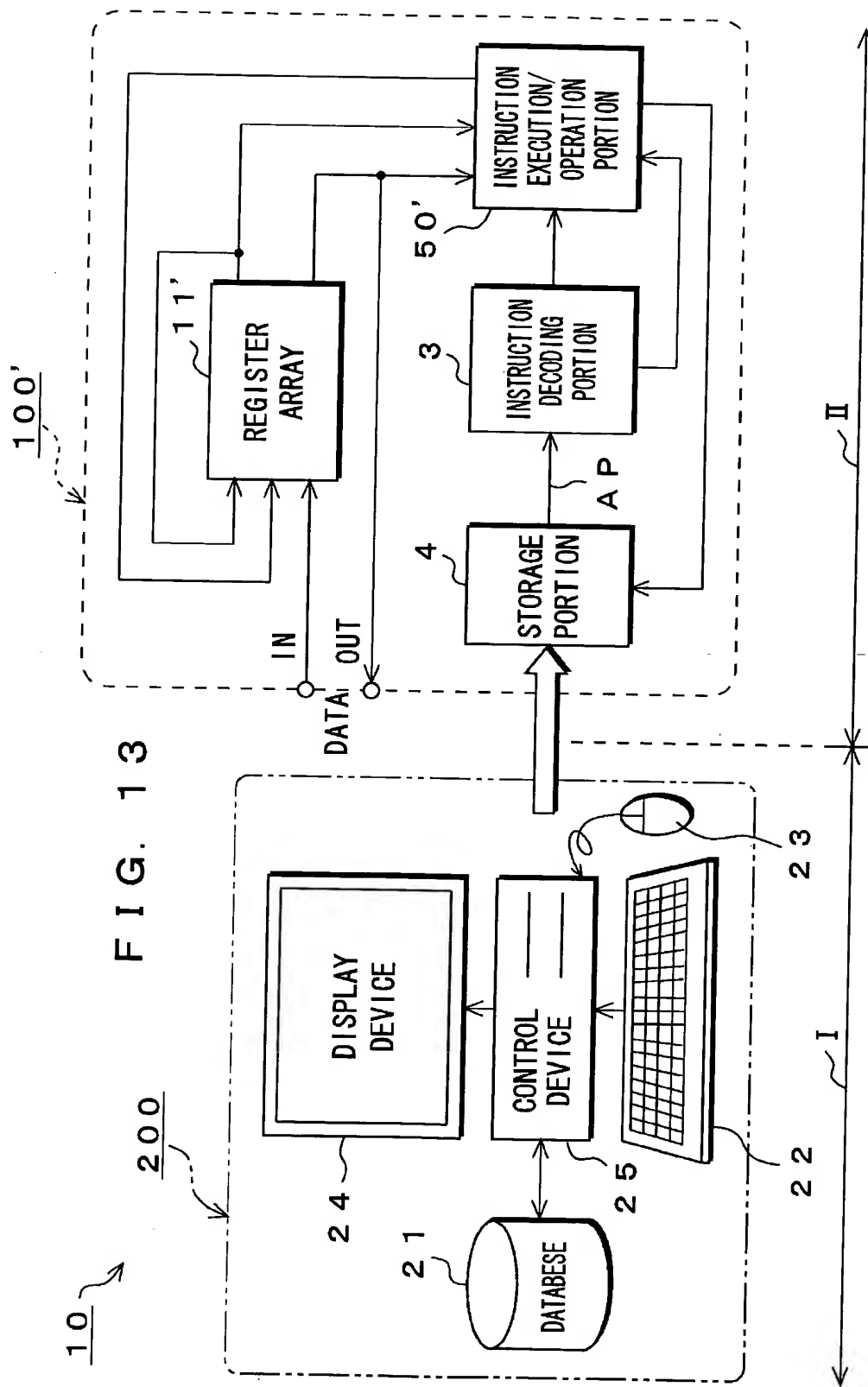


FIG. 13

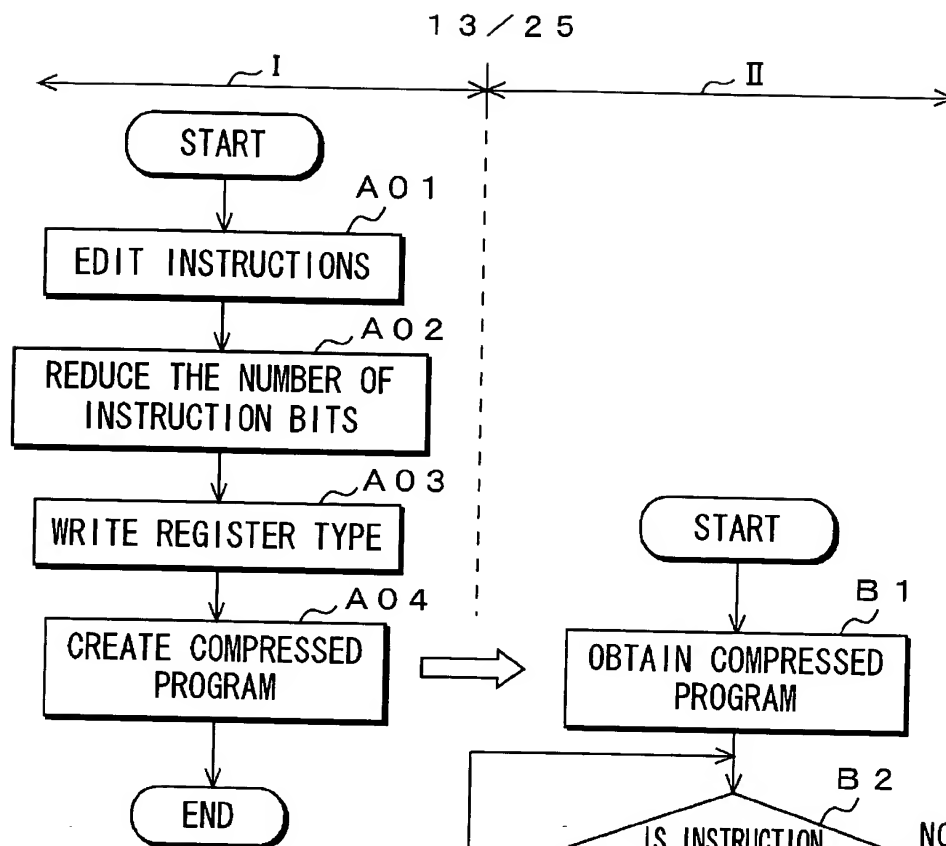


FIG. 14A

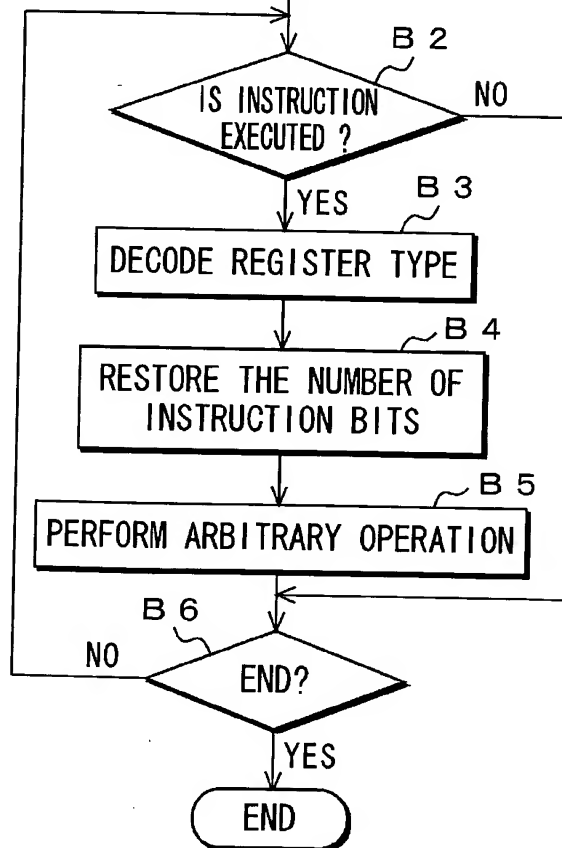
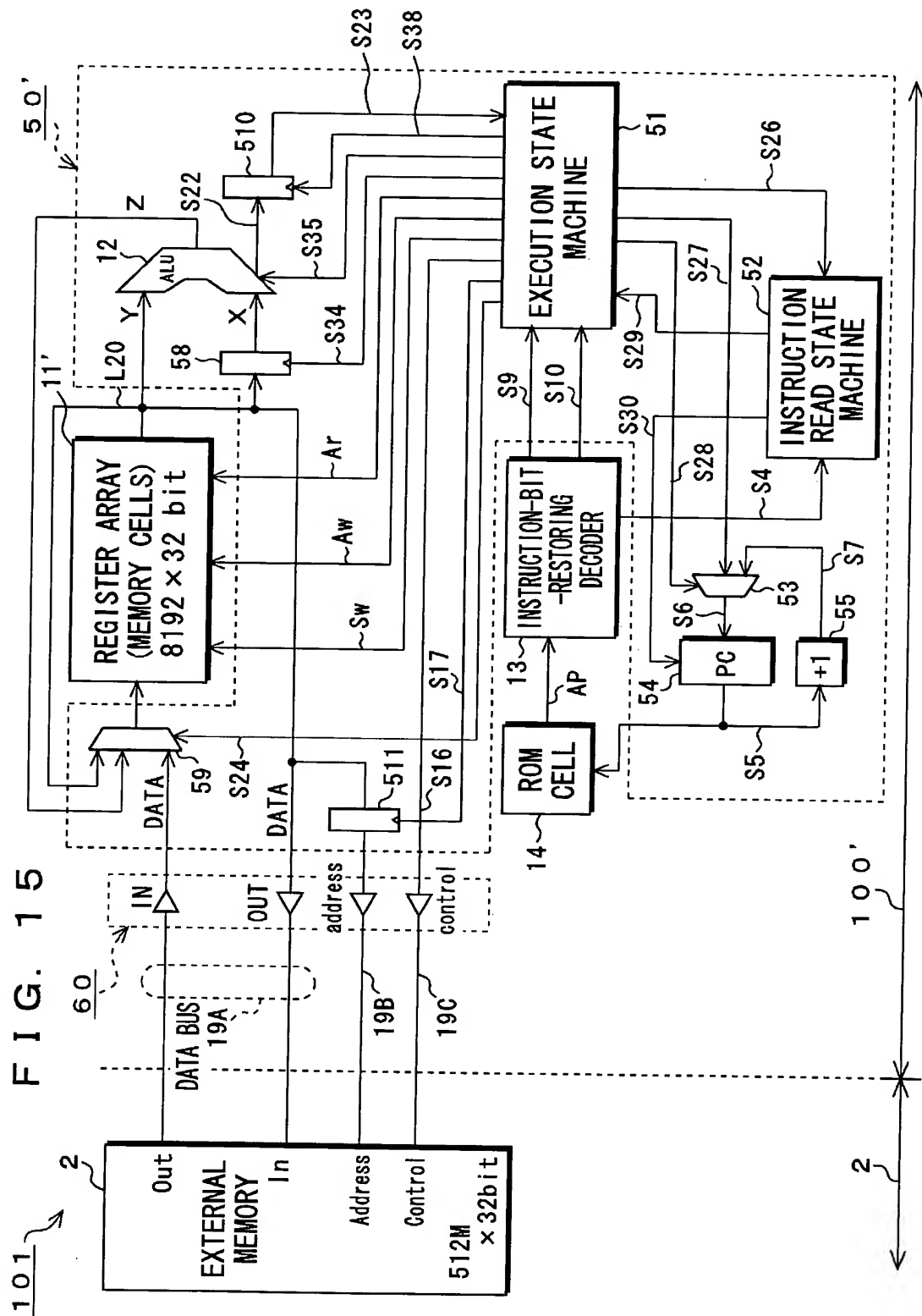


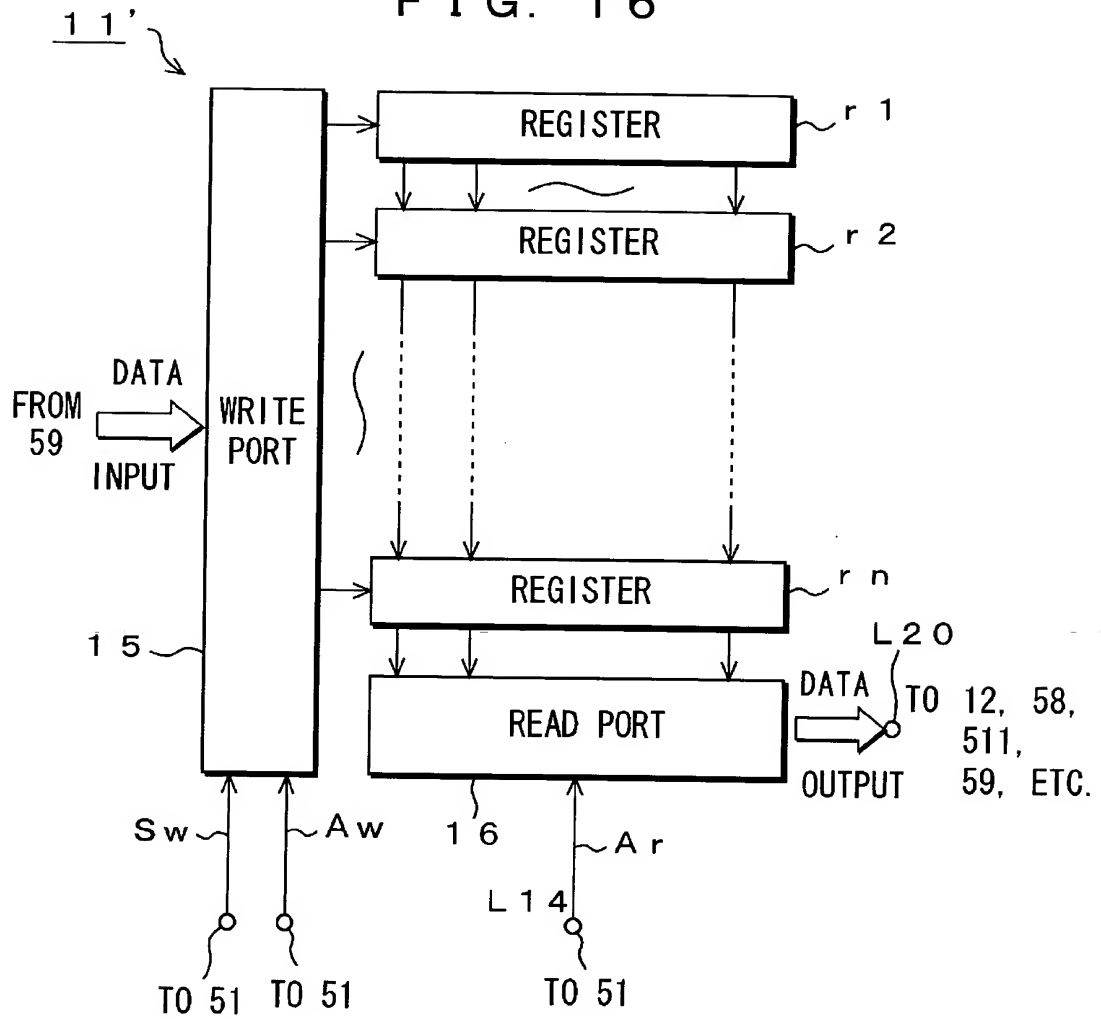
FIG. 14B

14/25



15/25

FIG. 16



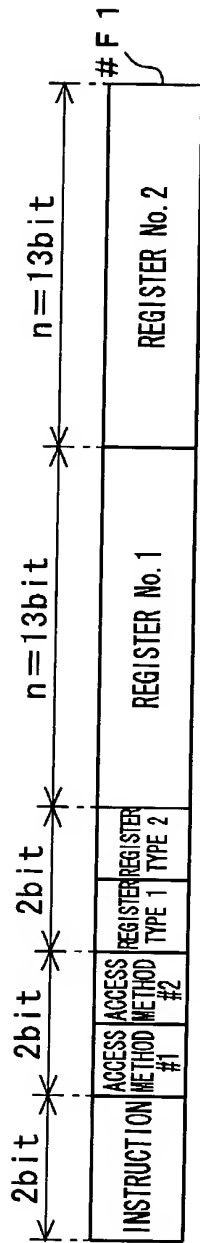


FIG. 17A load, add, OR
cmp INSTRUCTION

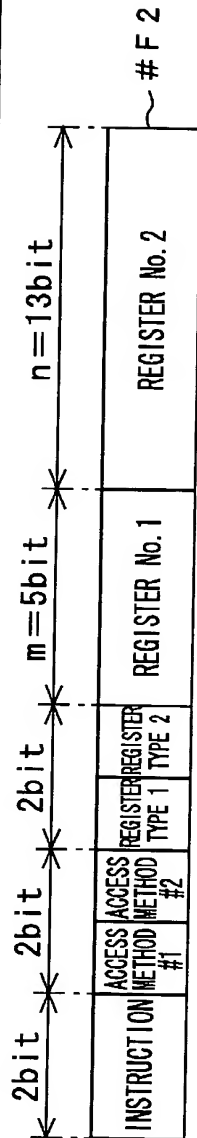


FIG. 17B load, add, OR
cmp INSTRUCTION

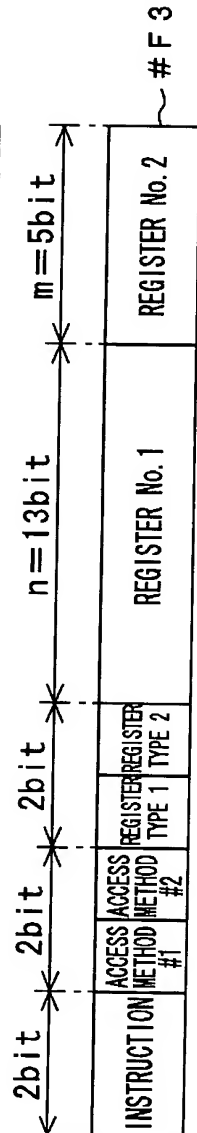


FIG. 17C load, add, OR
cmp INSTRUCTION

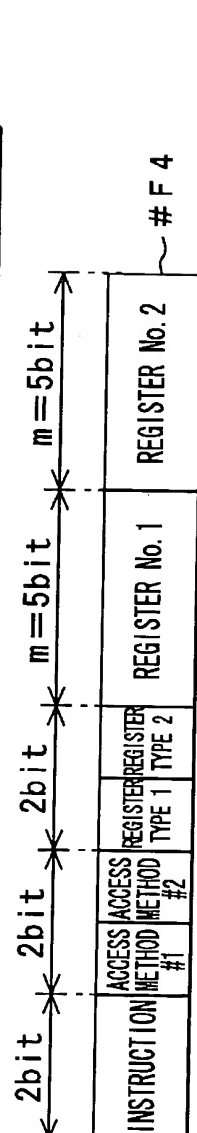


FIG. 17D load, add, OR
cmp INSTRUCTION

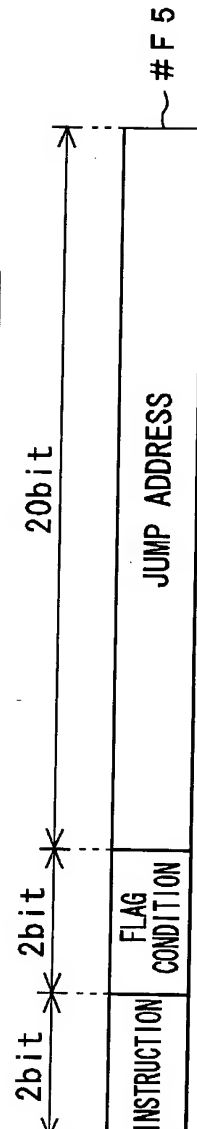


FIG. 17E jump
INSTRUCTION

17/25

INSTRUCTION	
0	load
1	add
2	cmp
3	jump

FIG. 18A

ACCESS METHOD #1, #2	
0	REGISTER, DIRECTLY
1	REGISTER, RELATIVELY


FIG. 18B

REGISTER TYPE 1, 2	
0	REGISTER No. OF 31 OR LOWER
1	REGISTER No. OF 32 OR HIGHER

FIG. 18C

FLAG CONDITION	
0	UNCONDITIONAL
1	zero flag
2	non-zero flag
3	NOT USED

FIG. 18D

I  FIG. 19

PROGRAM WRITTEN IN C LANGUAGE	WRITTEN CONTENTS	EXECUTION EXAMPLE
<pre>int *read_add, *write_add, counter, end_val; void main() { int temp, added_val; do { temp = *read_add; temp = temp + added_val; *write_add = temp; read_add = read_add + added_val; write_add = write_add + added_val; counter = counter + added_val; } while(counter != end_val); }</pre>	<p>Global VARIABLE DECLARATION</p> <p>FUNCTION DECLARATION</p> <p>Local VARIABLE DECLARATION</p> <p>SUBSTITUTION ADDITION SUBSTITUTION</p> <p>ADDITION ADDITION ADDITION</p> <p>COMPARISON, BRANCH-OFF</p>	<p>ASSIGNED TO 32'ND AND HIGHER-NUMBERED REGISTERS</p> <p>ASSIGNED TO 31'ST AND LOWER-NUMBERED REGISTERS</p>

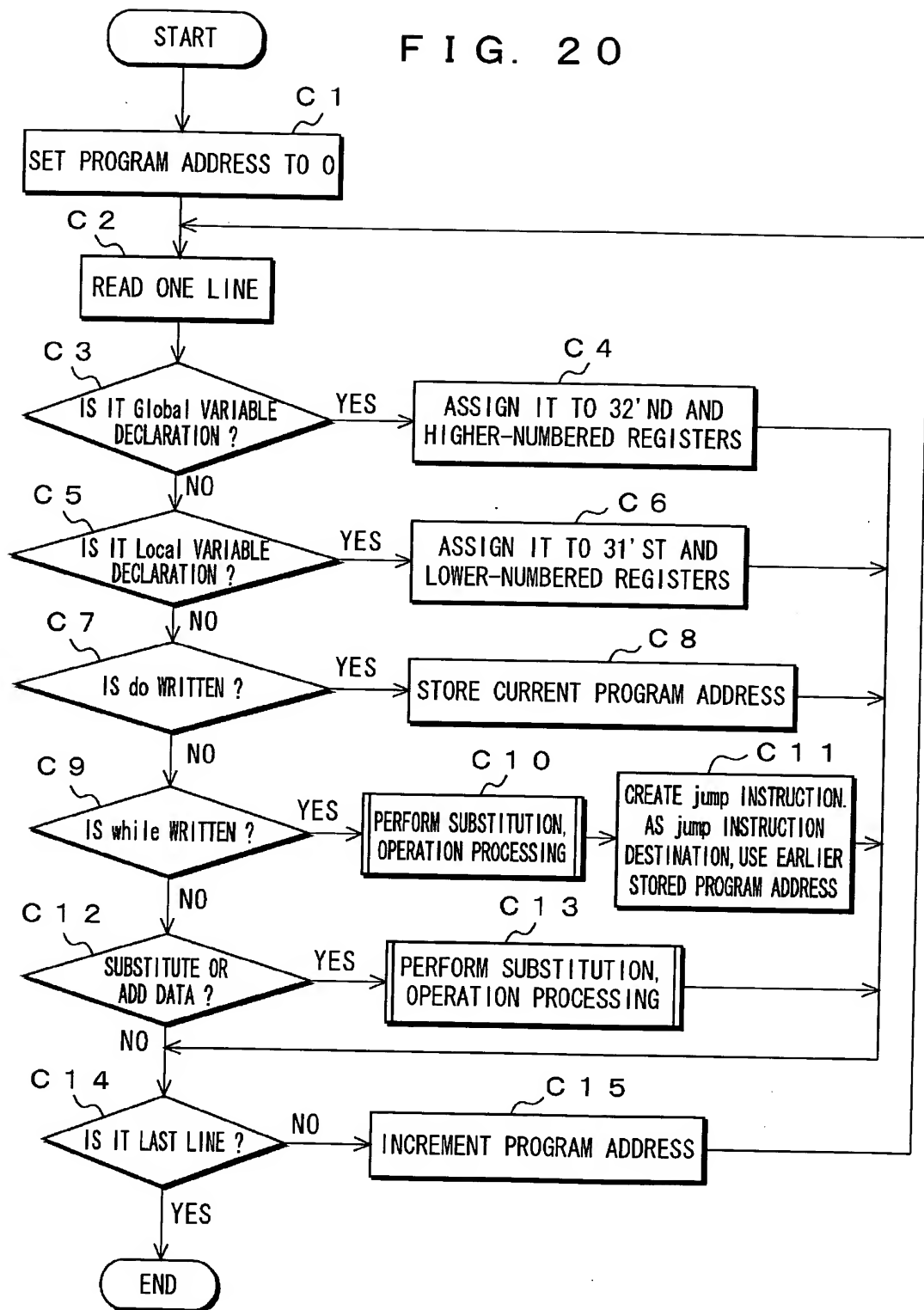
P 1

P 2

P 3

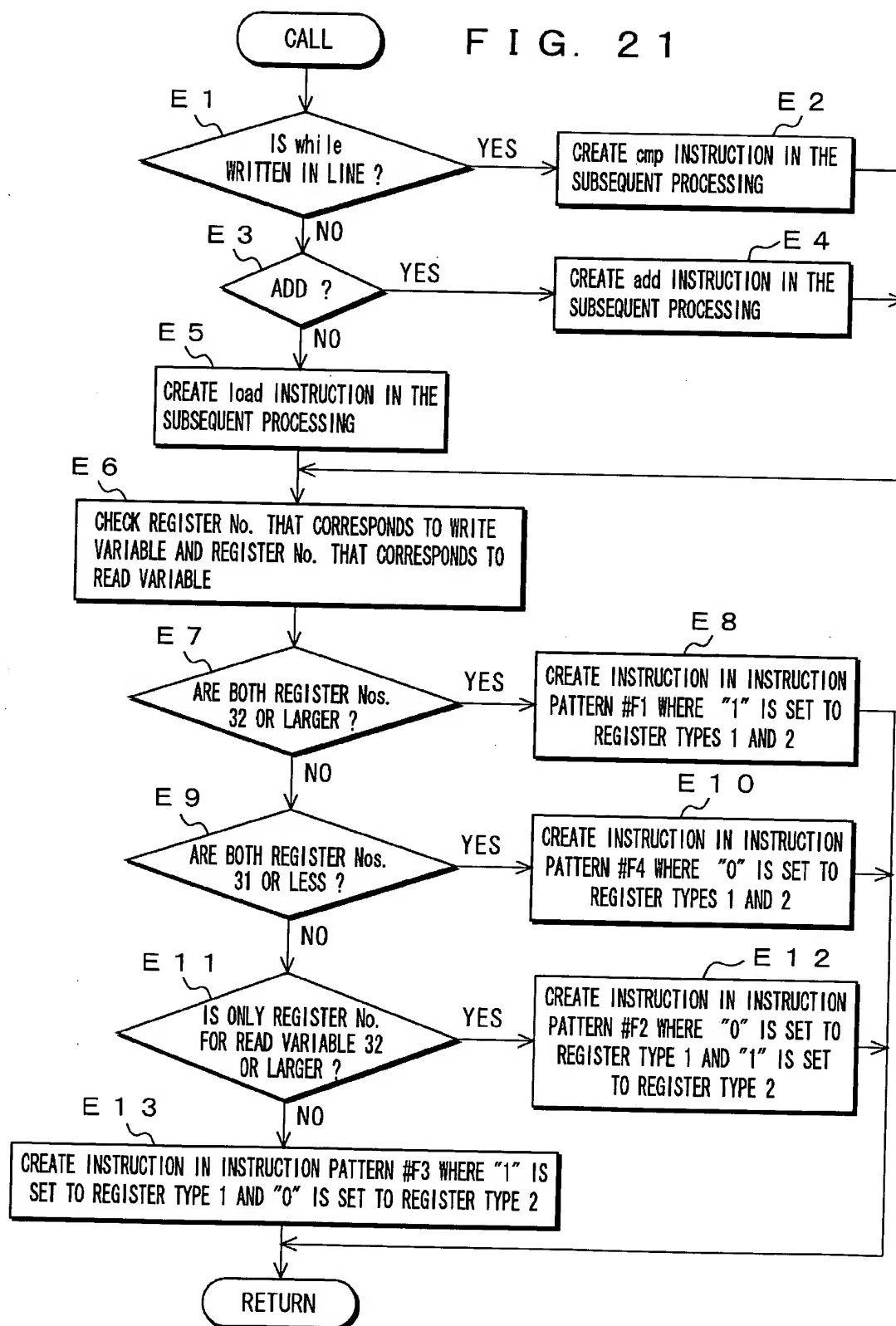
19/25

FIG. 20



20/25

FIG. 21



21/25

FIG. 22

No.	EXPRESSION BY MNEMONICS		EXPRESSION BY MACHINE LANGUAGE
# I 1	LOOP	load r 0, (r 32)	140020h
# I 2		add r 0, r 1	4001h
# I 3		load(r 33), r 0	280420h
# I 4		add r 32, r 1	480401h
# I 5		add r 33, r 1	480421h
# I 6		add r 34, r 1	480441h
# I 7		cmp r 34, r 35	8C044023h
# I 8		jump nz, LOOP	E00000h

FIG. 23

REGISTER No.	VALUE	
r 0	—	← TEMPORARILY USED REGISTER
r 1	1	← ADDED
r 3 2	0	← READ ADDRESS
r 3 3	1 0	← WRITE ADDRESS
r 3 4	0	← COUNTER
r 3 5	1 0	← END VALUE

22 / 25

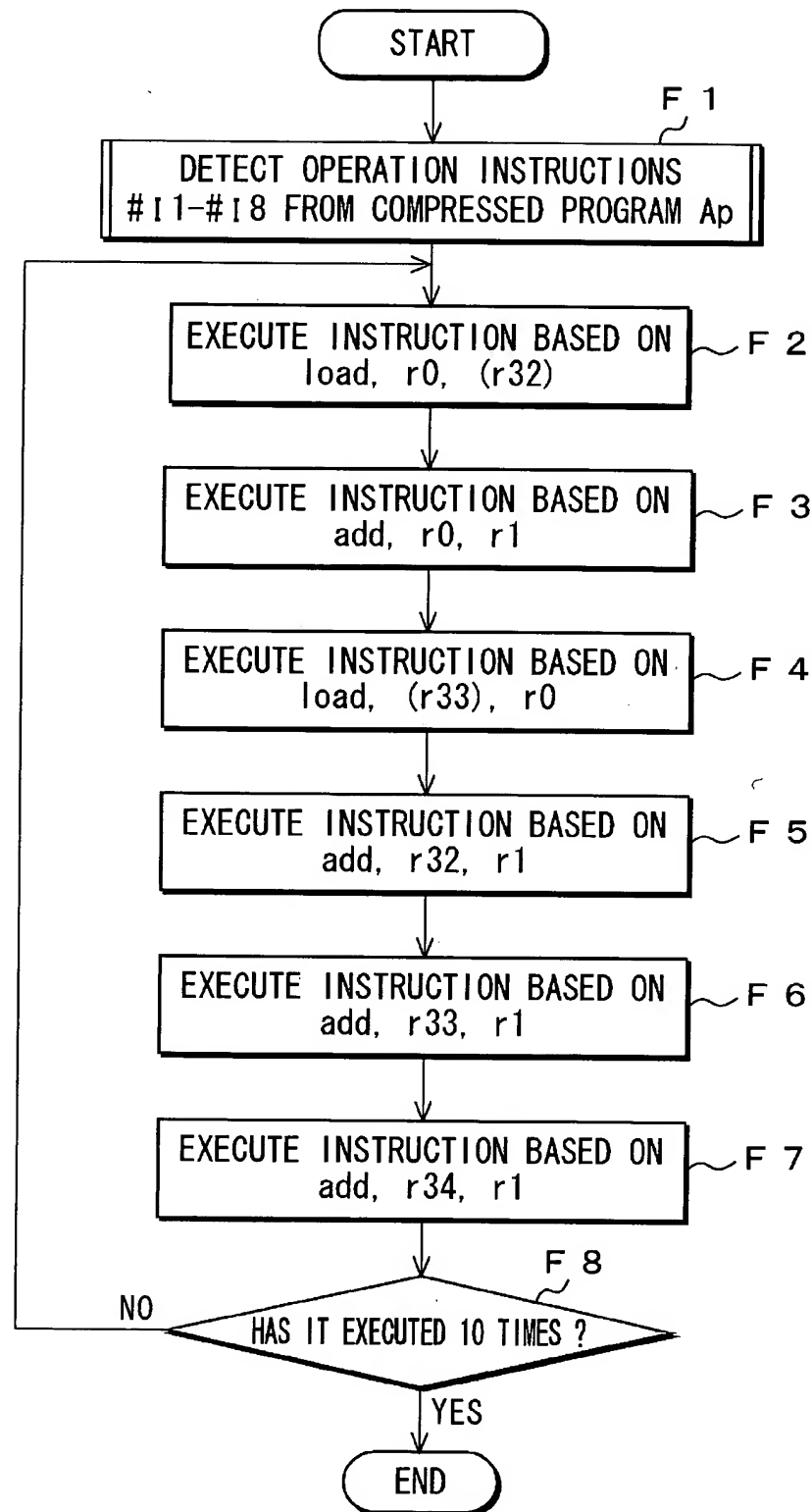
FIG. 24

2 ↘

ADDRESS	DATA	
0 0 0 0 h	0	↕ # M 1
0 0 0 1 h	0	
0 0 0 2 h	0	
0 0 0 3 h	0	
0 0 0 4 h	0	
0 0 0 5 h	0	
0 0 0 6 h	0	
0 0 0 7 h	0	
0 0 0 8 h	0	
0 0 0 9 h	0	
0 0 0 A h	0	↕ # M 2
0 0 0 B h	0	
0 0 0 C h	0	
0 0 0 D h	0	
0 0 0 E h	0	
0 0 0 F h	0	
0 0 1 0 h	0	
0 0 1 1 h	0	
0 0 1 2 h	0	
0 0 1 3 h	0	

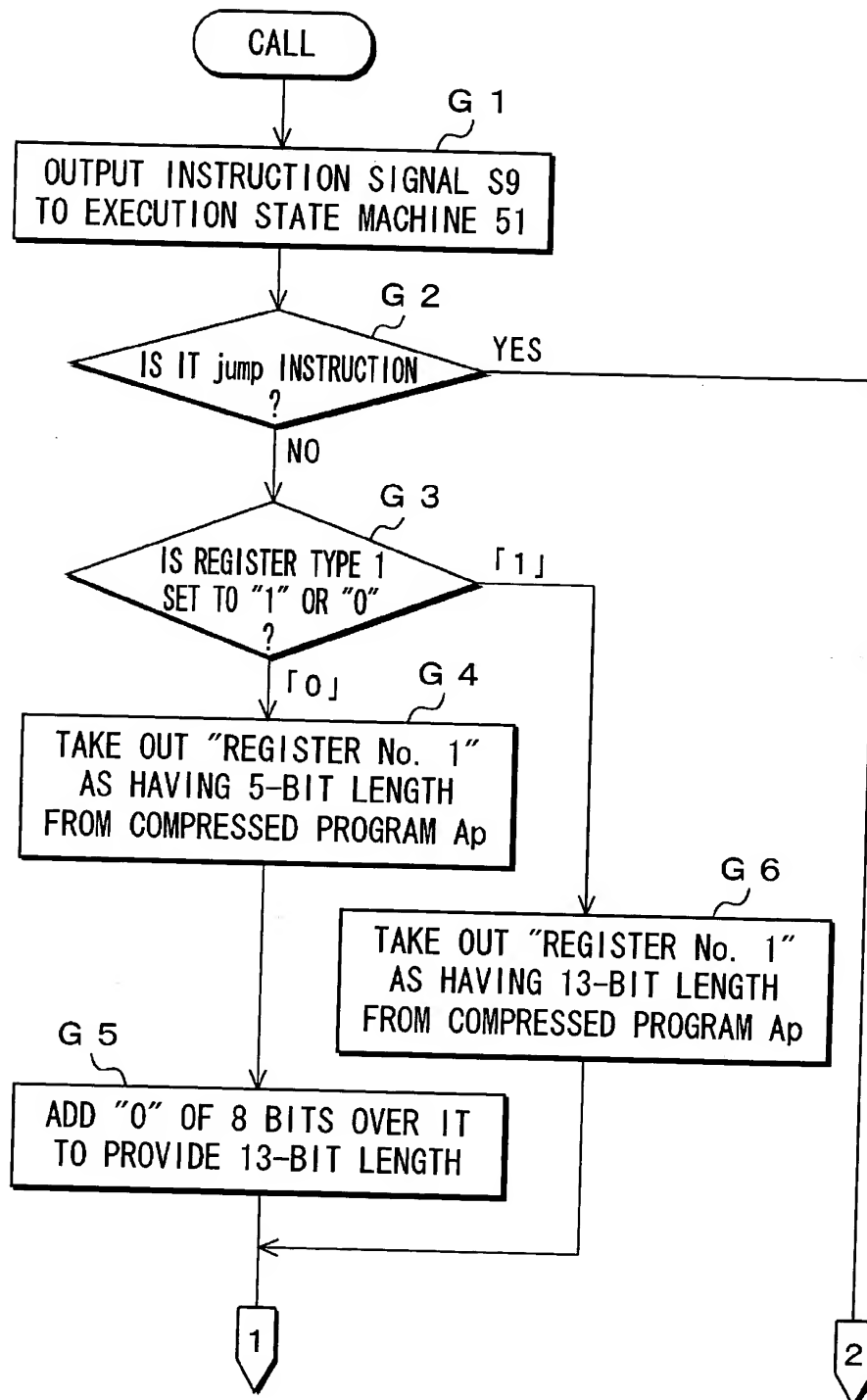
23 / 25

FIG. 25



24 / 25

FIG. 26A



25 / 25

FIG. 26B

